

Friend or Foe? Fake Profile Identification in Online Social Networks

Michael Fire, Dima Kagan, Aviad Elyashar, and Yuval Elovici*

Telekom Innovation Laboratories at Ben-Gurion University of the Negev
Department of Information Systems Engineering, Ben Gurion University

Abstract

The amount of personal information unwillingly exposed by users on online social networks is staggering, as shown in recent research. Moreover, recent reports indicate that these networks are infested with tens of millions of fake users profiles, which may jeopardize the users' security and privacy. To identify fake users in such networks and to improve users' security and privacy, we developed the Social Privacy Protector software for Facebook. This software contains three protection layers, which improve user privacy by implementing different methods. The software first identifies a user's friends who might pose a threat and then restricts this "friend's" exposure to the user's personal information. The second layer is an expansion of Facebook's basic privacy settings based on different types of social network usage profiles. The third layer alerts users about the number of installed applications on their Facebook profile, which have access to their private information. An initial version of the Social Privacy Protection software received high media coverage, and more than 3,000 users from more than twenty countries have installed the software, out of which 527 used the software to restrict more than nine thousand friends. In addition, we estimate that more than a hundred users accepted the software's recommendations and removed at least 1,792 Facebook applications from their profiles. By analyzing the unique dataset obtained by the software in combination with machine learning techniques, we developed classifiers, which are able to predict which Facebook profiles have high probabilities of being fake and therefore, threaten the user's well-being. Moreover, in this study, we present statistics on users' privacy settings and statistics of the number of applications installed on Facebook profiles. Both statistics are obtained by the Social Privacy Protector software. These statistics alarmingly demonstrate how exposed Facebook users information is to both fake profile attacks and third party Facebook applications.

Keywords. Social Network Security and Privacy, Fake Profiles, Online Social Networks, Facebook, Supervised Learning, Facebook Application, Facebook Friends Statistics, Facebook Applications Statistics, Facebook Users Privacy Settings.

1 Introduction

In recent years, online social networks have grown rapidly and today offer individuals endless possibilities for publicly expressing themselves, communicating with friends, and sharing information with people across the world. A recent survey [28] estimated that 65% of adult internet users use online social network sites, such as Twitter [40], LinkedIn [26], Google+ [18], and Facebook [9]. As of October 2012, the Facebook social network has more than one billion active users monthly [13]. On average, Facebook users have 138 friends and upload more than 219 billion pictures onto Facebook [13]. Moreover, according to the Nielsen "Social Media Report" [31], American Internet users spent more than 53.5 billion minutes on Facebook in the month of May 2011, making Facebook the leading web-brand in the United-States.

Due to the friendly nature of Facebook, users tend to disclose many personal details about themselves and about their connections. These details can include date of birth, personal pictures, work place, email address, high school name, relationship statuses, and even phone numbers. Moreover,

*Email: {mickyfi, kagandi, aviade, elovicig@bgu.ac.il}

Bosmaf et al. [4], discovered that an average of 80% of studied Facebook users accepted friend requests from people they did not know if they shared more than 11 mutual friends. In many cases, accepting friend requests from strangers may result in the exposure of a user’s personal information to third parties. Additionally, personal information of Facebook users can be exposed to third party Facebook applications [8]. Another privacy concern deals with existing privacy settings, for which the majority of Facebook users do not match the security expectations [27]. These results indicate that many users accidentally or unknowingly publish private information leaving them more exposed than they assumed.

If a user’s personal information is disclosed to a malicious third party, it can be used to threaten the well-being of the user both online and in the real world. For example, a malicious user can use the gained personal information and send customized spam messages to a user in an attempt to lure such users onto malicious websites [39], or blackmail them into transferring money to the attacker’s account [30]. To cover their tracks, social network attackers may use fake profiles. In fact, the number of fake profiles on Facebook can be counted in the tens of millions. According to a recent report [12], Facebook estimates that 8.7% (83.09 million), of its accounts do not belong to real profiles. Moreover, Facebook also estimates that 1.5% (14.32 million), of its accounts are “undesirable accounts”, which belong to users who may deliberately spread undesirable content, such as spam messages and malicious links, and threaten the security and privacy of other Facebook users.

In this study, we present the Social Privacy Protector software for protecting user privacy on Facebook (otherwise referred to as SPP). The SPP software consists of two main parts, namely, a Firefox add-on and a Facebook application. The two parts provide Facebook users with three different layers of protection. The first layer, which is part of the Firefox add-on, enables Facebook users to easily control their profile privacy settings by simply choosing the most suitable profile privacy settings with just one click. The second layer, which is also part of the software Firefox add-on, notifies users of the number of applications installed on their profile which may impose a threat to their privacy. The third layer, a Facebook application, analyzes a user’s friends list. By using simple heuristics (see Section 4.1), the application identifies which friends of a user are suspected as fake profiles and therefore impose a threat on the user’s privacy. The application presents a convenient method for restricting the access of fake profiles to a user’s personal information without removing them from the user’s friends list.

At the end of June 2012, we launched an initial version of the SPP software as a “free to use software” [14, 15] and received massive media coverage with hundreds of online articles and interviews in leading blogs and news websites, such as Fox news [3] and NBC news [32]. Due to the media coverage, in less than four months, 3,017 users from more than twenty countries installed the SPP Facebook application, 527 of which used the SPP Facebook application to restrict 9,005 friends¹. Moreover, at least 1,676 users installed the Firefox add-on out of which we estimate that 111 users used the add-on recommendation and removed more than 1,792 Facebook applications from their profiles (see Section 5.1). In addition, the add-on also succeeded in collecting the Facebook privacy settings of 67 different Facebook users.

To our great surprise many of the SPP application users used the application to not only remove users that were recommended for removal, but to also manually search and restrict them by name, specific friends that have a higher likelihood of having profiles that belong to real people. The removal of real profiles also assists us in studying and constructing classifiers that identify real profiles recommended for restriction. The collected data obtained from SPP users gave us a unique opportunity to learn more about user privacy on online social networks in general, and on Facebook in particular. Also, by using the unique data obtained from users restricting their Facebook friends, as well as by implementing machine learning techniques, we developed classifiers, which can identify user’s friends that are recommended for restriction (see Section 4.2). Our classifiers presented an AUC of up to 0.948, precision at 200 of up to 98%, and an average users precision at 10 of up to 24% (see Section 5). Furthermore, these types of classifiers can also be used by online social network administrators to

¹Due to the unexpected massive downloads and usage of the application our servers did not succeeded in supporting the large amount of users at once. Moreover, in our initial version, the SPP Facebook application did not support all the existing web browsers. Therefore, many users who installed the SPP software did not have the possibility to use it on demand.

identify and remove fake profiles from the online social network.

In this study we also present statistics on Facebook user privacy settings, which were obtained by the SPP Add-on. These statistics demonstrate how exposed Facebook users' personal information is to fake profile attacks and third party applications (see Section 5.3). For example, we show that out of 1,676 examined Facebook users, 10.68% have more than a hundred Facebook applications installed on their profile, and 30.31% of the users have at least 40 Facebook applications installed. Moreover, out of 67 collected users' privacy settings the majority of the user's personal information is set up to be exposed to friends leaving the user's personal information exposed to fake friends.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of various related solutions, which better help protect the security and privacy of social network users. In addition, we also present an overview of similar studies, which used machine learning techniques to predict user properties, such as predicting users' links in social networks. In Section 3, we describe the SPP software architecture in detail. In Section 4, we describe the methods and experiments used in this study. We describe the initial deployment of the SPP software and the methods used for the construction and evaluation of our machine learning classifiers. In Section 5, we present the results of our study, which include an evaluation of our classifiers and different users' privacy statistics obtained by the SPP software. In Section 6, we discuss the obtained results. Lastly, in Section 7, we present our conclusions from this study and offer future research directions.

2 Related Work

2.1 Online Social Network Security and Privacy

In recent years, due to the increasing number of privacy and security threats on online social network users, social network operators, security companies, and academic researchers have proposed various solutions to increase the security and privacy of social network users.

Social network operators attempt to better protect their users by adding authentication processes to ensure that a registered user represents a real live person [22]. Many social network operators, like Facebook, also offer their users a configurable user privacy setting that enables users to secure their personal data from other users in the network [27, 29]. Additional protection may include a shield against hackers, spammers, socialbots, identity cloning, phishing, and many other threats. For example, Facebook users have an option to report users in the network who harass other users in the network [11]. In addition, Facebook also developed and deployed an Immune System, which aims to protect its user from different online threats [38].

Many commercial and open source products, such as Checkpoint's SocialGuard [44], Websense's Defensio [7], UnitedParents [41], RecalimPrivacy [36], and PrivAware application [33], offer online social network users tools for better protecting themselves. For example, the Websense's Defensio software aims to protect its users from spammers, adult content, and malicious scripts on Facebook.

In recent years, several published academic studies have proposed solutions for various social network threats. DeBarr and Wechsler [6] used the graph centrality measure to identify spammers. Wang [42] presented techniques to classify spammers on Twitter based on content and graph features. Stringhini et al. [39] presented a solution for detecting spammers in social networks by using "honey-profiles". Egele et al. [8] presented PoX, an extension for Facebook, which makes all requests for private data explicit to the user. Yang et al. [43] presented a method to identify fake profiles by analyzing different features, such as links' creation timestamps, and friend requests frequency. Anwar and Fong [2] presented the Reflective Policy Assessment tool, which aids users in examining their profiles from the viewpoint of another user in the network. Rahman et al. [34] presented the MyPageKeeper Facebook application, which aims to protect Facebook users from damaging posts on the user's Facebook wall. In a later study [35], Rahman et al. also presented the FRAppE application for detecting malicious applications on Facebook. They discovered that 13% of one hundred and eleven thousand Facebook applications in their dataset were malicious applications. Recently, Fire et al. [16] proposed a method for detecting fake profiles in online social networks based on anomalies in a fake user's social structure.

In this study, we present the the SPP software, which offers methods for improving Facebook user

privacy². By using data collected by the SPP software and machine learning techniques, we present methods for constructing classifiers that can assist in identifying fake profiles.

2.2 Online Social Networks and Machine Learning

With the increasing popularity of social networks many researchers had studied and used a combination of data obtained from social networks and machine learning techniques to predict different user properties [25, 1, 37]. Furthermore, several studies used machine learning techniques to improve user security in online social networks [39, 23, 16].

In 2007, Liben-Nowell and Kleinberg [25] used machine learning techniques to predict links between users in different social networks (also referred to as the link prediction problem). In 2010, Stringhini et al. [39] proposed a method for detecting spammer profiles by using supervised learning algorithms. In the same year, Lee et al. [23] used machine learning and honeypots to uncover spammers in MySpace and Twitter. Sakaki et al. [37] used machine learning and content data analysis of Twitter users in order to detect events, such as earthquakes and typhoons, in real-time. In 2012, Altshuler et al. [1] used machine learning techniques to predict different user's properties, such as origin and ethnicity, inside the "Friends and Family" social network, which was created by logs extracted from a user's mobile device. Recently, Fire et al. [16] used the online social network's topological features to identify fake users in different online social networks.

As part of this study we present a method for recommending to a Facebook user which of his friends might be a fake profile and should therefore, be restricted. Our method is based on the connection properties between a Facebook user and its friends and by using supervised learning techniques. This type of problem is to some degree similar to the problem of predicting link strength, studied by Kahanda and Nevill [21], and the problem of predicting positive and negative links (signed links), as Leskovec et al. [24] studied. Similarly to the study held by Kahanda and Nevill, in this study we extract a different set of meta-content features, such as the number of pictures and videos both the user and his friends were tagged in. In this study, we also predict the type of a negative relationship between users, similar to the study of Leskovec et al. However, in our study we aim to uncover fake profiles rather than identify the link sign or strength between two users. In addition, our study contrasts other studies, which used a major part of the social network topology to construct classifiers [21, 24, 16], because we construct our classifiers by using only variations of the data collected in real-time from the user's point of view rather than data collected from the social network administrator's point of view. By using user data, which was obtained in real-time only, we were able to quickly analyze each user's friends list with fewer resources and without invading the user's friend's privacy.

3 Social Privacy Protector Architecture

To better protect the privacy of Facebook users we developed the *Social Privacy Protector* software. The SPP software consists of three main parts (see Figure 1), which work in synergy: a) *Friends Analyzer Facebook application* - which is responsible for identifying a user's friends who may pose a threat to the users privacy, b) *SPP Firefox Add-on* - which analyzes the user's privacy settings and assists the user in improving privacy settings with just one click, and c) *HTTP Server* - which is responsible for analyzing, storing, and caching software results for each user. In the remainder of this section, we describe in detail each individual part of the SPP software.

3.1 The Friends Analyzer Facebook Application

The Friends Analyzer Facebook application (also referred to as SPP application) is the part of the SPP responsible for analyzing a user friends list to determine which of the user's friends may pose a threat to the user's privacy. After the user installs the Friends Analyzer application, the application scans the user's friends list and returns a credibility score for each one of the user's friends. Each friend's score is created by using simple heuristics or with a more sophisticated machine learning algorithm,

²An initial version of the SPP software was described, as work in progress, in our previous paper [15]

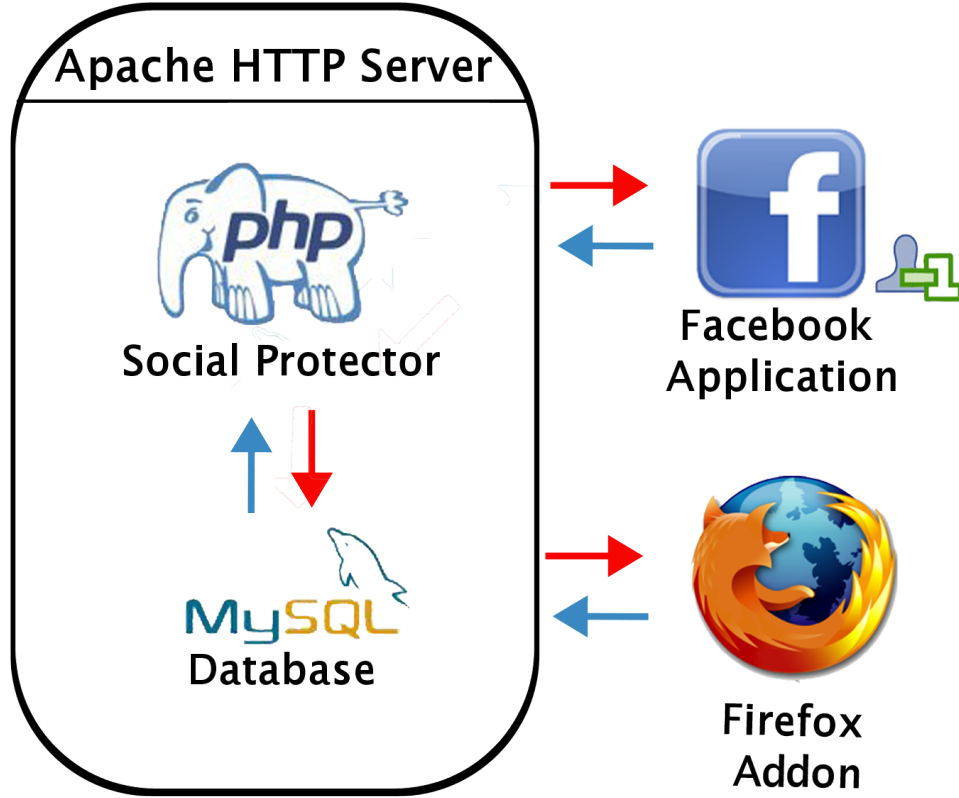


Figure 1: Social Privacy Protector Architecture.

which takes into account the strength of the connection between the user and his friends. The strength of each connection is based on different connection features, such as the number of common friends between the user and his friend and the number of pictures and videos the user and his friend were tagged in together (see Section 4). At the end of the process, the user receives a web page, which includes a sorted list of all his friends according to each friend's score, where the friends with the lowest scores have the highest likelihood of being fake profiles appear on the top of the list (see Figure 2). For each friend in the returned sorted list, the user has the ability to restrict the friend's access to the user's private information by simply clicking on the restrict button attached to each friend in the sorted list. Moreover, the application provides the user an interface to view all his friends alphabetically and easily restricts access with a single click. This option enables Facebook users to protect their privacy not only from fake profiles but also from real profiles, such as exes, for whom they do not want to have access to their personal data stored in their Facebook profile.

3.2 Social Privacy Protector Firefox Add-on

The Social Privacy Protector Firefox Add-on (also referred to as Add-on) is the part of the SPP software responsible for improving user privacy settings with just a few simple clicks. After the Add-on is installed on the user's Firefox browser, it begins to monitor the user's internet activity. When the Add-on identifies that the user logged onto his Facebook account, the Add-on then analyzes the number of applications installed on the user's Facebook profile and presents a warning with the number of installed applications, which may pose a threat to the user's privacy (see Figure 3). The Add-on also presents the top two results obtained by the Friends Analyzer Facebook application and suggests which friends to restrict (see Figure 3).

The Add-on also detects when the user has entered Facebook's privacy settings page and presents the user with three new privacy setting options. The new privacy settings are based on the user's profile type and can be modified with one click (see Figure 4), instead of the more complex Facebook

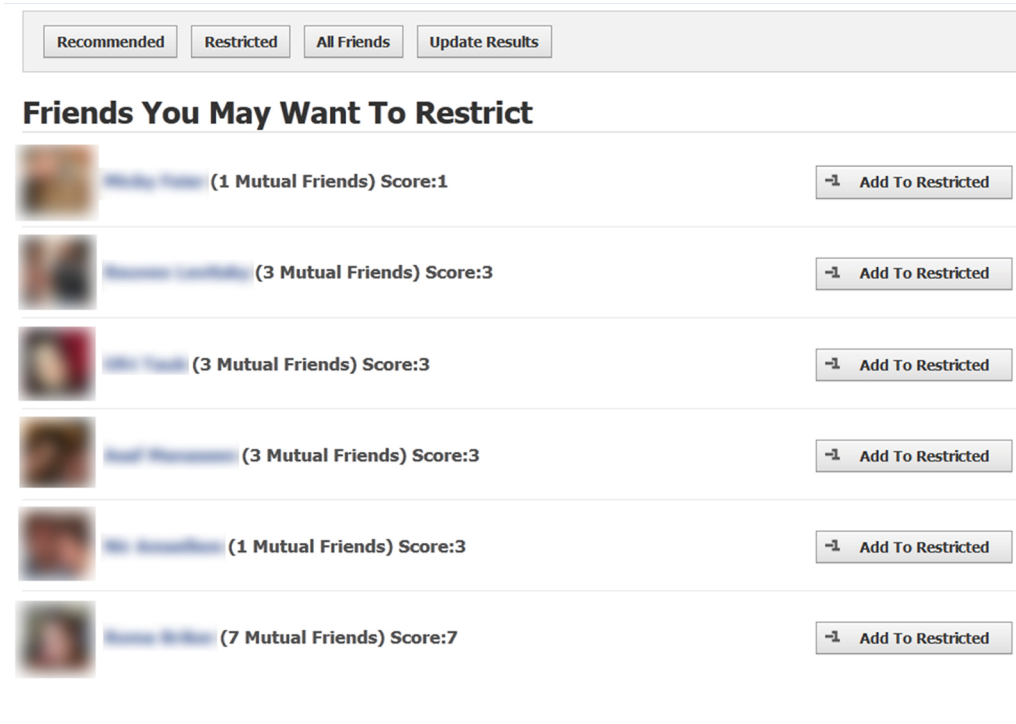


Figure 2: Friends Analyzer Facebook application - user rank and sorted friends list.

custom privacy settings that may contain up to 170 options [2]. Using the new Add-on privacy settings a user can simply chose the profile type most suitable for him out of three options: a) *Celebrity setting* - in this setting all of the user's information is public, b) *Recommended setting* - in this setting the user's privacy is only public to friends, however some of the user's details, such as profile name and pictures, are public, and c) *Kids settings* - in this setting the profile is only open to the user's friends and only friends of friends can apply for friend requests. Using this Add-on a user can easily control and improve their privacy without contacting a security expert. Moreover, parents can simply install this Add-on on their children's Facebook accounts in order to better protect their children's privacy on Facebook without needing to understand Facebook's different privacy setting options. Our Add-on is also easy for customizing privacy settings by adding more privacy option settings to different types of users. Furthermore, it is easy to customize our privacy settings by adding more optional privacy settings for different types of users. In this study, we utilized users data collected by the Add-on to further study the privacy settings of Facebook users.

3.3 HTTP Server

The HTTP server is the part of the SPP responsible for connecting the SPP Firefox Add-on to the SPP Facebook application. When a user installs the SPP software, the server analyzes the user's friends list and identifies which of the user's friends may pose a threat on his security. Also, to enhance the application's performance, the HTTP server caches parts of the analyzed results. In order to protect the user's privacy, the application stores only the minimal number of features in an encrypted manner using RC4 encryption.

4 Methods and Experiments

In this study our experiments are divided into two main parts. In the first part, we deployed an initial version of the SPP software in order to improve user privacy on Facebook. We also used the initial version to collect data on each SPP's user and his links. The main focus of this part is calculating the heuristic, which sorts the friends list and recommends which friends to restrict. Additionally, in

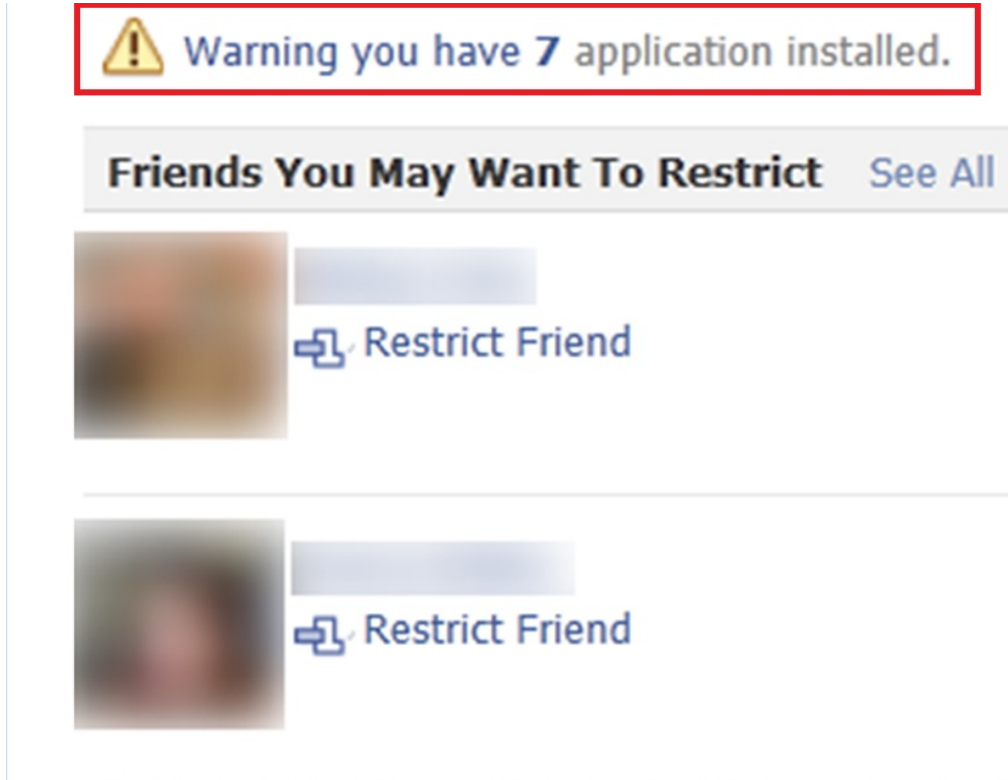


Figure 3: Social Privacy Protector Firefox Add-on - warning about installed applications and friends you may want to restrict.

this part, we also present methods for analyzing the user privacy setting data collected by the Add-on to better understand Facebook user privacy settings and understand how much they are exposed to different security threats, such as information exposure to fake friends.

In the second part, we use the data collected in the first part to learn more about Facebook user's privacy settings. Furthermore, we used the collected data in order to develop machine learning classifiers, which can identify Facebook profiles with higher likelihoods of being fake. Moreover, these classifiers can be used to replace the initial SPP heuristic with a more generic model, which can provide SPP users recommendations on which friends to restrict. In the second part our main focus is on constructing the machine learning classifiers and evaluating their performances.

In the remainder of this section, we present in detail the methods used in each one of the two parts.

4.1 Deploying Social Privacy Protector - Initial Version

After we developed the SPP's software according to the architecture described in Section 3, we had to develop a heuristic, which can quickly sort the friends list of each SPP user. In the initial version, we developed the heuristic to be as simple as possible and based it upon the hypothesis that most fake users do not have strong connections with real users. To estimate the strength of a connection between two users, we extracted lists of features and calculated a simple arithmetic heuristic. The heuristic's main constrain was that for every SPP user, it needed to analyze and evaluate hundreds or even thousands of connection strengths between the user and each one of his friends in a short period of time. Moreover, the heuristic needed to take into consideration the performance of the Facebook application API [10], in its calculation time of each feature. After testing and evaluating several different features, we decided to extract the following features for each SPP user (referred as u), and each one of his friends (referred as v):

1. **Are-Family(u,v)** - mark if u and v are defined in Facebook as being in the same family. The *Are-Family* feature prevents the cases in which the SPP application will mark a family member as a fake profile.

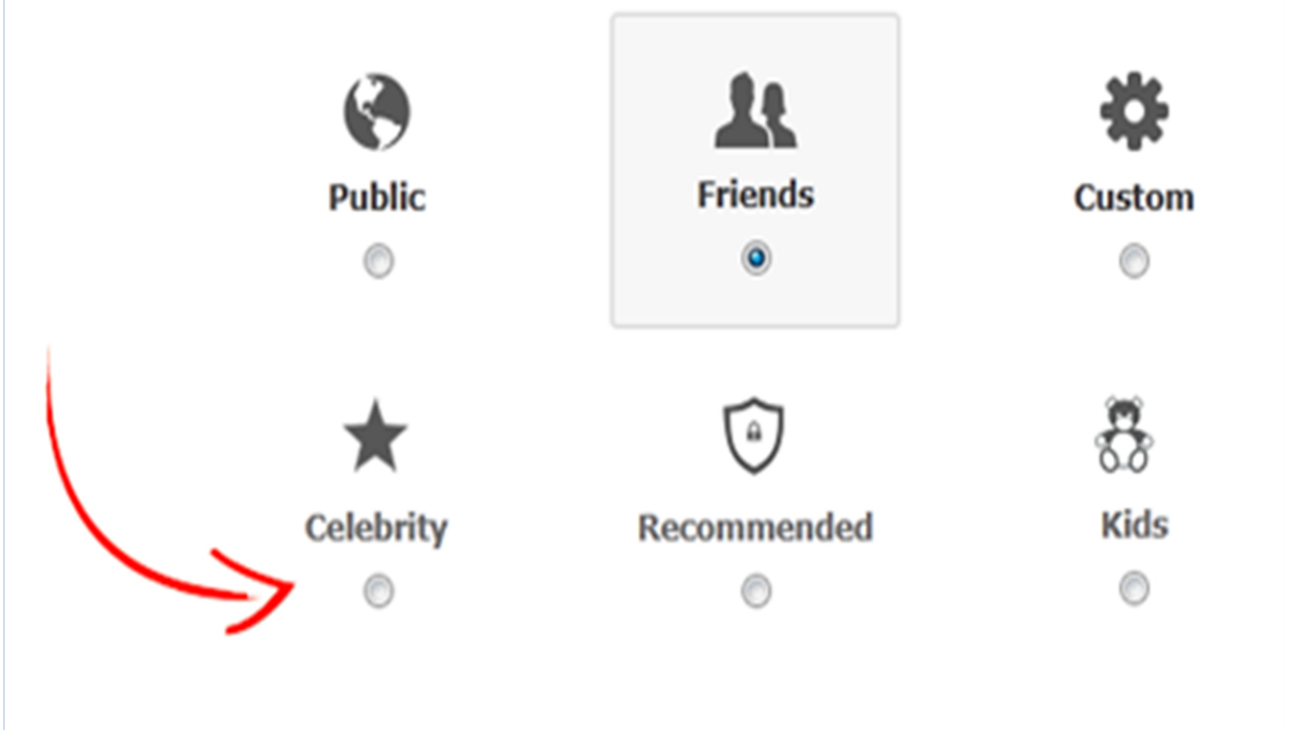


Figure 4: Social Privacy Protector Firefox Add-on - optimizing the user's privacy setting with one simple click.

2. **Common-Chat-Messages(u, v)** - the number of chat messages sent between u and v . We assume that in most cases, such as in a fake profile used to send spam messages, there will be no chat interaction between the user and the fake profile. However, when there are different types of fake profiles, such as fake profiles used by cyber predators, this feature will be less helpful in identifying the threat.
3. **Common-Friends(u, v)** - the number of mutual friends both u and v poses. The relevance of the *Common-friends* feature is very intuitive. It is expected that the larger the size of the common neighborhood, the higher the chances are that the friendship between the users is real. The Common-Friends feature was previously used to solve different versions of the link prediction problem [25, 21, 24, 17] and was found to be a very useful feature in many of these scenarios.
4. **Common-Groups-Number(u, v)** - the number of Facebook groups both u and v are members in. It is expected that the higher the number of groups both users are member of, the higher the chances that u and v have similar fields of interest, which might indicate that the friendship between u and v is real. The Common-Groups-Number feature was used in the study of Kahanda and Neville [21] to predict link strength.
5. **Common-Posts-Number(u, v)** - the number of posts both u and v posted on each other's wall in the last year. Similar post features were studied by Kahanda and Neville and were discovered to be very useful in predicting strong relationships between two Facebook users [21].
6. **Tagged-Photos-Number(u, v)** - the number of photos both u and v were tagged in together. We assume that most fake profiles have almost no shared tagged photos with the user. The Tagged-Photos-Number feature was also used in the Kahanda and Neville study [21].
7. **Tagged-Videos-Number(u, v)** - the number of video clips both u and v appeared in together. As in the case of tagged photos, we assume that most fake profiles have almost no shared tagged video clips with the user.

8. **Friends-Number(u) and Friends-Number(v)** - the total number of friends u and v have. These features are usually referred to as the *degree* of u and the *degree* of v , and were extracted from each user to assist us in improving the supervised learning classifiers, as described in Section 4.2. However, we did not use this feature when calculating the connection strength heuristics in SPP's initial version.

In an attempt to build better fake profile prediction heuristics, we also tested the following features: a) the number of mutual Facebook likes both u and v gave to each other, and b) the number of comments both u and v posted on each other's wall posts. However, although these two features seemed promising in assisting us in identifying fake profiles, we did not use them in the end due to performance issues, i.e., calculating these two features was too time consuming and inappropriate to use in order to receive real-time results.

After several tests and simple evaluations, we decided to use the following simple heuristic in order to define the *Connection-Strength* function between a user u and its friend v :

$$\begin{aligned} \text{Connection-Strength}(u, v) = CS(u, v) &:= \text{Common-Friends}(u, v) \\ &+ \text{Common-Chat-Messages}(u, v) \\ &+ 2 \cdot \text{Common-Groups-Number}(u, v) \\ &+ 2 \cdot \text{Common-Posts-Number}(u, v) \\ &+ 2 \cdot \text{Tagged-Photos-Number}(u, v) \\ &+ 2 \cdot \text{Tagged-Videos-Number}(u, v) \\ &+ 1000 \cdot \text{Are-Family}(u, v) \end{aligned}$$

In order to tell SPP users which friends ought to be restricted, we ranked each user's friends list according to the *Connection-Strength*(u, v) function. Due to Facebook's estimations that 8.7% of all Facebook users do not belong to real profiles [12], we presented to each SPP user the top 10% of his friends who received the lowest Connection-Strength score (see illustration in Figure 2).

To evaluate the performance of the Connection-Strength heuristic, we calculated three statistics on the heuristic's performances in restricting friends. First, we calculated the heuristic's restricting precision for different Connection-Strength values. This calculation was performed by measuring, for different Connection-Strength values, the ratio between the number of friends who were restricted and the total number of friends who received the exact Connection-Strength value. Second, we calculated the restriction rates according to the friends' ranking positions in the restriction interface. This calculation was performed by measuring the percentage of friends, which were restricted in each position in the restriction interface. Lastly, we also calculated the heuristic's average users precision at k for different k values, in the following manner. First, for each SPP user u , which had at least k friends, we calculated the user's Connection-Strength average precision at k . This was done by selecting k friends, which received the lowest Connection-Strength values with u out of all u 's friends³. We then calculated the ratio between the number of friends u had restricted among the selected friends and k . After we finished calculating the Connection-Strength average precision at k for each user u , we then continued to calculate heuristic's average users precision at k , by simply summing up all the users' Connection-Strength average precision at k , and dividing the sum by the number of users with at least k friends. A formal arithmetical definition of the heuristic's average users precision at k is as follows:

$$CS\text{-Avg-Precision}(k) := \frac{\sum_{\{u \in Users \mid |friends(u)| \geq k\}} P_u(k)}{|\{u \in Users \mid |friends(u)| \geq k\}|}$$

Where $Users$ is a set, which contains all SPP users, $friends(u)$ is a set which contains all of u 's Facebook friends, and $P_u(k)$ defined to be the heuristic's precision at k for a user u :

$$P_u(k) := \frac{\sum_{\{f \in friends(u) \mid \exists f_i, \dots, f_{n-k} \in friends(u), \forall j \in [i, \dots, n-k] CS(f) \geq CS(f_j)\} is-restricted(u, f)}}{k}.$$

³In case more than k friends received the lowest Connection-Strength values, we randomly removed friends with the highest Connection-Strength values, until we were left with exactly k friends.

Where $is-restricted(u, f)$ is a function, which returns 1 if a user u had restricted his friend f or 0 otherwise.

The goal of the presented Connection-Strength heuristic was not to identify fake profiles in an optimal way, rather to make Facebook users more aware of the existence of fake profiles and the fact that these types of profiles can threaten their privacy. Additionally, we wanted to collect a unique dataset that would contain labeled profiles with high likelihoods of being fake profiles.

In addition to collecting meta-content features through the SPP application, the SPP Firefox Add-on also collected the following users defined privacy settings each time the user used the Add-on:

1. **Installed-Application-Number** - the number of installed Facebook applications on the user's Facebook account.
2. **Default-Privacy-Settings** - the default user's privacy settings on Facebook, which can be one of these values: public, friends or custom. This setting is responsible for the scope in which content created by the user will be exposed by default. For example, if the default privacy settings are set to "friends" by default, only the user's friends can see his posted content. The SPP's Add-on only stored this value if the user's privacy was set to public or friends. This privacy setting default value for new Facebook users is set to public.
3. **Lookup** - regulates who can look up the user's Facebook profile by name. This setting can take one of the following values: everyone, friends or friends of friends. This privacy setting default value for new Facebook users is set to everyone.
4. **Share-Address** - this value is responsible for defining who can see the user's address. This setting can take one of the following values: everyone, friends or friends of friends. This privacy setting default value for new Facebook users is set to everyone.
5. **Send-Messages** - this value is responsible for defining who can send messages to the user. This setting can take one of the following values: everyone, friends or friends of friends. This privacy setting default value for new Facebook users is set to everyone.
6. **Receive-Friend-Requests** - this value is responsible for defining who can send friend request to the user. This setting is limited to two values only : everyone and friends of friends. The default value for this setting is everyone.
7. **Tag-Suggestions** - this value is responsible for defining which Facebook users will receive photo tag suggestions when photos that look like the user have been uploaded onto Facebook. This setting can take one of the following values: no one or friends. This privacy setting default value for new Facebook users is set to friends.
8. **View-Birthday** - this value is responsible for defining who can view the user's birthday. This setting can take one of the following values: everyone, friends or friends of friends. This privacy setting default value for new Facebook users is set to friends-of-friends.

By analyzing and monitoring the privacy settings, we can learn more about the SPP user's privacy settings on Facebook. In addition, we can estimate how vulnerable Facebook users' information is to fake profile attacks. Furthermore, by analyzing the collected privacy settings, we can also identify other potential privacy risks, which are common to many different users.

4.2 Supervised Learning

After, we deployed the SPP software and gathered enough data on which friends SPP users had restricted, and which friends they had not restricted, our next step was to use supervised learning techniques to construct fake profile identification classifiers. To construct the fake profile identification classifiers, we first needed to define the different datasets and their underlining features. Next, we used different supervised learning techniques to construct the classifiers. Lastly, we evaluated the classifiers using different evaluation methods and metrics.

In the remainder of this section we describe, in detail, the process of constructing and evaluating our classifiers. First, in Section 4.2.1, we describe how we defined the different datasets and their features. Second, in Section 4.2.2, we describe which methods were used to construct our classifiers and evaluate their performance.

4.2.1 Datasets and features

The SPP application’s initial version collected and calculated many different details about each connection between a SPP user and each one of his friends in real-time (see Section 4.1). Moreover, the SPP application presented the user with two interfaces for restricting his friends. The first restriction interface (referred to as the recommendation interface) presents the user with a list with the 10% of his friends who received the lowest Connection-Strength score. The second restriction interface (referred to as the alphabetical interface) presents the user with all of his friends in alphabetical order. Using these two restriction interfaces, we defined four types of links sets, two unrestricted links sets, and two restricted links sets:

1. **All unrestricted links set** - this set consists of all the links between the application users and their Facebook friends who were not restricted by the application.
2. **Recommended unrestricted links set** - this set consists of all the links between the application users and their Facebook friends who were recommended for restriction by the application due to a low Connection-Strength score, but who were not restricted by the user.
3. **Recommended restricted links set** - this set consists of all the links between the application users and their Facebook friends who were recommended for restriction by the application due to a low Connection-Strength score and who were restricted by the user.
4. **Alphabetically restricted links set** - this set consists of all the links between the application users and their Facebook friends who were not recommended for restriction by the application. However, the user deliberately chose to restrict them by using the alphabetical interface⁴.

Using the defined above links sets, we define the following three datasets:

1. **Fake profiles dataset** - this dataset contains all the links in the *Recommended unrestricted links set* and in the *All unrestricted links set*. Namely, this dataset contains all friends who were restricted due to a relatively low Connection-Strength and all friends who were not restricted. Therefore, we believe that this dataset is suitable for constructing classifiers, which can predict friends, who mostly represent fake profiles, the user need to restrict. We believe that this dataset is suitable for replacing the Connection-Strength heuristics with a generic classifier, which can recommend to a SPP user, which friends to restrict. In addition, the classifiers constructed from this type of dataset can assist online network administrators in identifying fake profiles across the entire the network.
2. **Friends restriction dataset** - this dataset contains all the links in the *alphabetically restricted links set*, and in the *All unrestricted links set*. Namely, this dataset contains all the friends who were not restricted and all the friends who were restricted deliberately by the user, although they were not recommended by the SPP application. Therefore, we believe that this dataset is suitable for constructing classifiers, which can predict friends, who mostly represent real profiles the user prefers to restrict.
3. **All links dataset** - this dataset contains all the links in all four disjoint links sets. According to the dataset definition, this dataset is the largest among all defined datasets, we believe that like the Fake profiles dataset this dataset can be suitable for replacing the Connection-Strength heuristics with a generic classifier, which can recommend to a SPP user which friends to restrict.

⁴If a restricted user’s friend was presented in the recommendation interface and was restricted by using the alphabetical interface, the link between the user and the restricted friend was assigned to the recommended restricted links set.

For each link in the above defined links datasets, the SPP application calculated all of the 8 first link features defined in Section 4.1 in real-time including the $\text{Friends-Number}(v)$ ⁵ (referred to as the *Friend Friends-Number*). In addition, if it was arithmetically possible, we also calculated the following set of seven features:

1. **Chat-Messages-Ratio(u,v)** - the ratio between the number of chat message u and v sent to each other and the the total number of chat messages u sent to all of his friends. The formal *Chat-Messages-Ratio* definition is:

$$\text{Chat-Messages-Ratio}(u, v) := \frac{\text{Common-Chat-Messages}(u, v)}{\sum_{f \in \text{friends}(u)} \text{Common-Chat-Messages}(u, f)}$$

Where $\text{friends}(u)$ is defined to be a set which contains all the friends of u .

2. **Common-Groups-ratio(u,v)** - the ratio between the number of Facebook groups both u and v have in common and the maximum number of groups which u and all of his friends have in common. The formal *Common-Groups-Ratio* is:

$$\text{Common-Groups - Ratio}(u, v) := \frac{\text{Common-Groups-Number}(u, v)}{\max(\{\text{Common-Groups-Number}(u, f) | f \in \text{friends}(u)\})}$$

3. **Common-Posts-Ratio(u,v)** - the ratio between the number of posts both u and v posted on each others walls and the total number of posts which u posted on all his friends' walls. The formal *Common-Posts-Ratio* is:

$$\text{Common-Posts-Ratio}(u, v) := \frac{\text{Common-Posts-Number}(u, v)}{\sum_{f \in \text{friends}(u)} \text{Common-Posts-Number}(u, f)}$$

4. **Common-Photos-Ratio(u,v)** - the ratio between the number of tagged photos both u and v were tagged in together and the total number of photos, which u were tagged in. The formal *Common-Photos-Ratio* is:

$$\text{Common-Photos-Ratio}(u, v) := \frac{\text{Common-Photos-Number}(u, v)}{\sum_{f \in \text{friends}(u)} \text{Common-Photos-Number}(u, f)}$$

5. **Common-Video-Ratio(u,v)** - the ratio between the number of videos both u and v were tagged in together on and the total number of videos, which u were tagged in. The formal *Common-Video-Ratio* is:

$$\text{Common-Video-Ratio}(u, v) := \frac{\text{Common-Video-Number}(u, v)}{\sum_{f \in \text{friends}(u)} \text{Common-Video-Number}(u, f)}$$

6. **Is-Friend-Profile-Private(v)** - in some cases the SPP application did not succeed in collecting v 's friends number ($\text{Friends-Number}(v)$), and succeeded in collecting the $\text{Common-Friends}(u, v)$ value, which returned a value greater than zero. This case may indicate that v 's profile is set to be a private profile. With these cases in mind, we defined the *Is-Friend-Profile-Private* function to be a binary function, which returns true values in case the application did not succeed in collecting v 's friends number and succeeded in collecting the $\text{Common-Friends}(u, v)$ with a value greater than zero, or a false value otherwise.

7. **Jaccard's-Coefficient(u,v)** - Jaccard's-coefficient is a well-known feature for link prediction [25, 21, 17]. The Jaccard's coefficient is defined as the number of common-friends u and v have divided by the sum of distinct friends both u and v have together. The formal *Jaccard's-Coefficient* definition is:

$$\text{Jaccard's - Coefficient}(u, v) := \frac{\text{Common-Friends}(u, v)}{\text{Friends-Number}(u) + \text{Friends-Number}(v) - \text{Common-Friends}(u, v)}$$

A higher value of *Jaccard's-coefficient* denotes a stronger link between two Facebook users.

⁵In some cases we were not able to extract the user's (v) friends number probably due to the v 's privacy settings.

4.2.2 Classifiers Construction and Evaluation

Using the three datasets and the 15 features defined in the previous sections, we constructed classifiers for fake profile identification and for recommending profiles for restriction.

The process of constructing and evaluating the different classifiers was as follows. First we matched the suitable datasets for each type of classification mission in the following manner: a) for identifying fake profiles we used *Fake profiles dataset*, b) for recommending real profiles for restriction we used the *Friends restriction dataset*, and c) for replacing recommending to SPP users which real and fake friends to restrict we used the *All links dataset*. Next, for each link in each one of the datasets, we extracted the 15 features defined in Sections 4.1 and 4.2.1, and created a vector set for each link. Furthermore, we added an additional binary target feature that indicates if the link between the SPP user and his friend was restricted by the SPP user to each link’s features vector set. Due to the fact that the majority of the links in each dataset had not been restricted, these datasets were overwhelmingly imbalanced with restricted links as a minority class. Therefore, a naive algorithm that always predicts “not restricted”, will present good prediction precision. To overcome the datasets imbalance problem we used a similar undersampling methodology used by Guha et al. to predict trust [19] and by Leskovec et al. [24] to predict positive and negative links. According, to this methodology we transform each imbalanced dataset into a balanced dataset by combing all the restricted links in each dataset and adding to them an equal number of randomly selected unrestricted links from each dataset. Afterwards, we used the balanced datasets with the updated extracted links’ features vector sets to construct several classifiers by using WEKA [20], a popular suite of machine learning software written in Java and developed at the University of Waikato, New Zealand. We used WEKA’s C4.5 (J48), IBk, NaiveBayes, Bagging, AdaBoostM1, RotationForest, and RandomForest implementations of the corresponding algorithms. In addition, we used the simple OneR classifier as a baseline for the performance of the other classifiers. For each of these algorithms, most of the configurable parameters were set to their default values with the following exceptions: for C4.5, the minimum number of instances per leaf parameter was between the values of 2, 6, 8 and 10; for IBk, its k parameter was set to 5 and 10; The ensemble methods were configured as follows: The number of iterations for all ensemble methods was set to 100. The Bagging, AdaBoostM1, and RotationForest algorithms were evaluated using J48 as the base classifier with the number of instances per leaf set to 4, 6, 8, and 10. Next, we evaluated our classifiers using the common 10-folds cross validation approach. We used the area-under-curve (AUC), f-measure, true-positive and false-positive rate to evaluate the different classifiers’ performances. Additionally, in order to obtain an indication of the usefulness of the various features, we also analyzed the features importance by using WEKA’s information gain attribute selection algorithm.

Furthermore, in order to evaluate the classifiers recommendations precision at top k (*precision@k*), we selected the machine learning algorithm, which presented the highest AUC in the above evaluations and used two evaluation methods to measure the performance of the algorithm on the different datasets. In the first evaluation method, we split our datasets into training sets and testing sets. For each one of the three balanced datasets, we randomly split each dataset into a training dataset, which contained $\frac{2}{3}$ of the labeled instances and a testing dataset, which contained $\frac{1}{3}$ of the labeled instances. Afterwards, we constructed classifiers using the training dataset only. Next, we used the classifiers to classify the profiles in the testing dataset and sorted the instances according to the classifiers’ prediction probabilities in descending order, where the links, which received the highest probability of being restricted were first. We then evaluated the classifiers’ predictions precisions for the top k predictions, for different values of k .

In the second evaluation method, our goal was to measure the classifiers recommendations average users precision at k . To achieve this goal we used the following method. First, we selected a user out of all SPP users. Afterwards, we created a training dataset using all SPP users’ links without the selected user’s links. Next, we balanced the training dataset using the same undersampling method described above. Afterwards, we constructed a classifier using the training dataset and used the selected user’s links as a testing dataset. We used the constructed classifier to predict the probability of restriction for each link in the selected user’s links. We then sorted the classifier’s predictions in descending order where the links, which received the highest probability of being restricted were first. Subsequently, we measured the classifier’s predictions precision for different k values. Lastly, we repeated this process

for each one of all SPP users and calculated the average classifiers' precisions for different k values.

Using these evaluation methods, we were able to evaluate how precise our classifiers are in recommending which friends to restrict both from the SPP users' point of view, and from the online social network administrator point of view.

5 Results

The initial version of the SPP software was formally launched at the end of June 2012 as free to use software [14, 15]. The software launch received massive media coverage with hundreds of online articles and interviews in leading blogs and news websites, such as Fox news [3] and NBC news [32].

Due to the media coverage, from the 27th of June, 2012 to the 10th of November, 2012, 3,017 users from more than twenty countries installed the SPP application out of which 527 users used the SPP application to restrict 9,005 friends, with at least one friend restricted for each user. In addition, more than 1,676 users had installed the SPP Firefox Add-on and removed at least 1,792 applications.

In the remainder of this section we present the results obtained from analyzing the collected SPP software data in the following manner. First, in Section 5.1, we present the datasets obtained by the SPP application. Afterwards, in Section 5.2, we present the results obtained by our machine learning classifiers. Lastly, in Section 5.3 we present statistics on the success of our Add-on to assist Facebook users in removing unneeded applications from their profiles. Furthermore, in this section, we also presented different statistics about Facebook user privacy settings obtained from the SPP Add-on.

5.1 Collected Datasets

After the initial software launch the SPP application was installed by 3,017 users out of which 527 users had restricted 9,005 friends. All friends were restricted between the 27th of June, 2012 and the 10th of November, 2012. To our great surprise 355 SPP application users used the application not only to remove users that received low Connection-Strength score, but to also search and restrict specific friends that are probably real profiles by name.

Using the collected users' data we created three datasets as described in Section 4.2.1 (see Table 1). The first dataset was the Fake-profiles dataset, this dataset contained 141,146 out of which the 434 SPP users had restricted 2,860 links (2.03% of all links), which were recommended by the SPP application. The second dataset was the Friends-restriction dataset, this dataset contained 144,431 links out of which the 355 users had restricted 6,145 links (4.25% of all links), that were specifically chosen for restriction by the users using the Alphabetical-interface. The last dataset was the All links dataset, which contained 151,825 links out of which 9,005 (6.01% of all links), were restricted. As expected all three datasets were overwhelmingly imbalanced with imbalance rates ranging from 2.03% to 6.01%.

Table 1: Links Datasets

	Users Number	Restricted Link	Unrestricted Links	Total Links
Fake-Profiles	434	2,860 (2.03%)	138,286	141,146
Friends Restriction	355	6,145 (4.25%)	138,286	144,431
All Links	527	9,005 (6.01%)	138,286	147,291

To better understand the differences between the restricted links features and unrestricted links features, we calculated the average values of each extracted feature in each dataset for each link type (see Table 2). It can be noted that in all the examined features, except for the *Friend Friends-Number* and *Is-Friend-Profile-Private* features, the restricted links features received a lower average than the unrestricted links features in each dataset.

To understand how well the Connection-Strength heuristic performed, we calculated, as described in Section 4.1, the heuristic's restricting precision for different Connection-Strength values (see Figure 5), the heuristic restriction rates according to the friends' ranking positions in the Restriction interface (See Figure 6), and the heuristic's average users precision at k for different values of k (see Figure 7).

Table 2: Features Average Values for Different Datasets

Feature	Link Type	Fake Profiles	Friends Restriction	All Links
Are-Family	Restricted	0	1 link	1 link
	Unrestricted	9 links	9 links	9 links
Common-Chat-Messages	Restricted	0.02	6.35	4.34
	Unrestricted	30.86	30.86	30.86
Common-Friends	Restricted	1.44	19.8	13.97
	Unrestricted	36.78	36.78	36.78
Common-Groups-Number	Restricted	0.028	0.56	0.392
	Unrestricted	0.689	0.689	0.689
Common-Posts-Number	Restricted	0.008	0.069	0.049
	Unrestricted	0.147	0.147	0.147
Tagged-Photos-Number	Restricted	0.004	0.208	0.143
	Unrestricted	0.3	0.3	0.3
Tagged-Videos-Number	Restricted	0	0.007	0.005
	Unrestricted	0.017	0.017	0.017
Friend Friends-Number	Restricted	627.31	819.57	756.25
	Unrestricted	703.31	703.31	703.31
Chat-Message-Ratio	Restricted	$2.46 \cdot 10^{-5}$	0.003	0.002
	Unrestricted	0.004	0.004	0.004
Common-Groups-Ratio	Restricted	0.006	0.108	0.076
	Unrestricted	0.118	0.118	0.118
Common-Posts-Ratio	Restricted	0.0003	0.003	0.002
	Unrestricted	0.0035	0.0035	0.0035
Common-Photos-Ratio	Restricted	$2.23 \cdot 10^{-5}$	0.003	0.002
	Unrestricted	0.0034	0.0034	0.0034
Common-Video-Ratio	Restricted	0	0.001	0.0007
	Unrestricted	0.0027	0.0027	0.0027
Is-Friend-Profile-Private	Restricted	5.87%	10.79%	9.23%
	Unrestricted	9.81%	9.81%	9.81%
Jaccard's-Coefficient	Restricted	0.003	0.034	0.024
	Unrestricted	0.045	0.045	0.045

Although the Connection-Strength heuristic was quite simple, it presented an average users precision of 33.6% at 1, an average users precision of 27.1% at 10, and an average users precision of 11% at 100 (see Figure 7). In addition, 31.7% of the friends, which appeared in the second position in the Restriction interface, due to a low Connection-Strength score, were actually restricted by the SPP users (See Figure 6). Furthermore, 28% of the SPP users' friends who received a Connection-Strength of 0 were also restricted (see Figure 5). However, the friends' restriction rates sharply declined when the Connection-Strength score increased. For example, only 10%, of the users' friends, which received a Connection-Strength equal to 3 were actually restricted.

5.2 Classifiers' Results

From the three imbalanced datasets, we created three balanced datasets by using all the restricted links in each dataset and randomly choosing an equal amount of unrestricted links. We then used the balanced dataset and evaluated the specified machine learning algorithms (see Section 4.2.2) using a 10-fold cross-validation approach. The evaluation results of the different classifiers are presented in Figure 8 and in Table 3. It can be seen that on all datasets the Rotation-Forest classification algorithm presented the best AUC results among all the ensemble classifiers and the J48 decision tree classification algorithm presented the best results among all the non-ensemble classifiers. In addition,

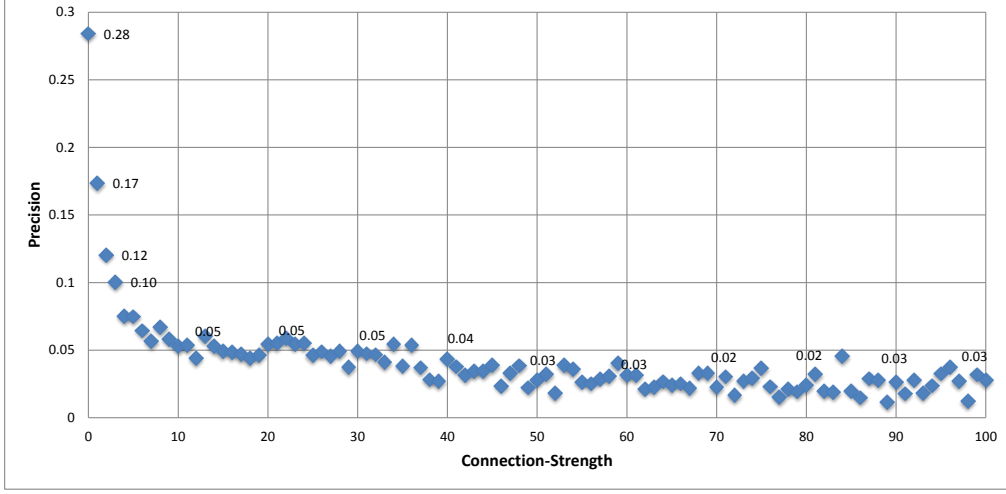


Figure 5: Friends restriction precision for different Connection-Strength values - it can be noted that among all users' friends, which received a Connection-Strength of 3, only 10% were actually restricted.

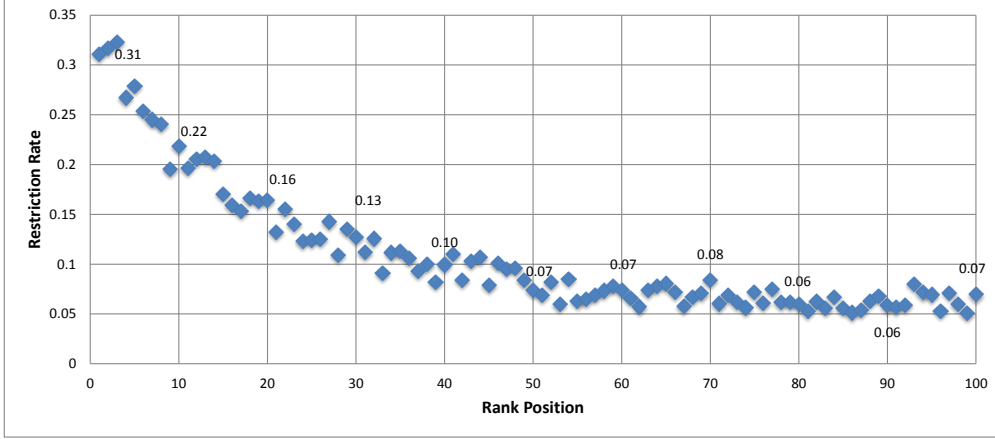


Figure 6: Connection-Strength restriction rates according to friends' ranking positions in the Restriction interface - it can be noted that among all friends, which were ranked in the first position in the friends Restriction interface 31.1% were actually restricted by the SPP users.

it can be noted that on all datasets the Rotation-Forest classifier presented considerably better results than the simple OneR classifier, which we used as baseline.

After, we discovered that the Rotation-Forest classifier presented the best overall results, we evaluated the Rotation-Forest classifier precision at k for different values of k on the different datasets. We first calculated the classifier precision for different k values by splitting each dataset into a training dataset, which contained $\frac{2}{3}$ of the links and a testing dataset, which contained $\frac{1}{3}$ of the links. The results of this precision at k evaluation are presented in Figure 9. It can be noted that the Rotation-Forest classifiers presented precision at 200 of 98%, 93%, and 90% for the Friends restriction dataset, Fake profiles datasets, and All links dataset respectively. In addition, it can be noted that the classifiers' precision at 500 was 94%, 91%, and 88% for the Fake profiles datasets, Friends restriction dataset, and All links dataset. Hence, out of 500 links, which ranked were by the the Rotation-Forest classifiers as links with the highest likelihood of being restrict by the SPP application users, 470, 455, and 440 links were actually restricted in the Fake profiles datasets, Friends restriction dataset, and All links dataset respectively.

In order to estimate the classifiers' recommendations precision according to the SPP users' point

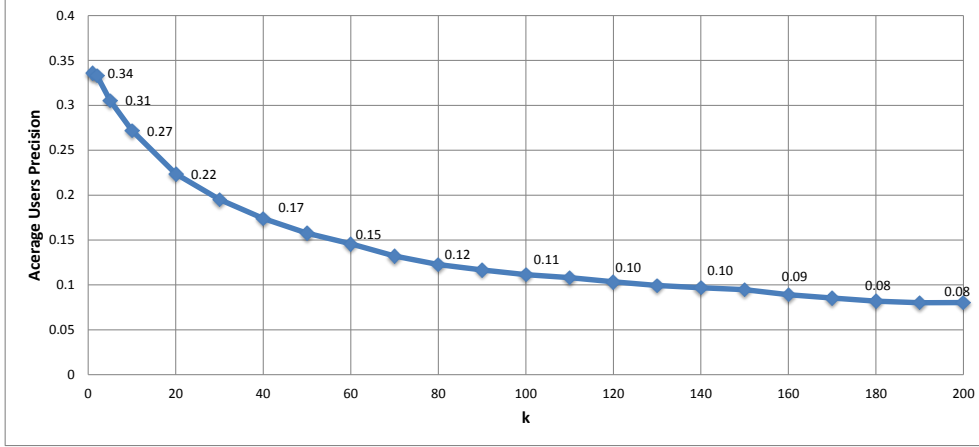


Figure 7: Connection-Strength average users precision at k - it can be noted that the Heuristic's average users precision at 1 and average users precision at 100 was 33.6% and 11.1% respectively.

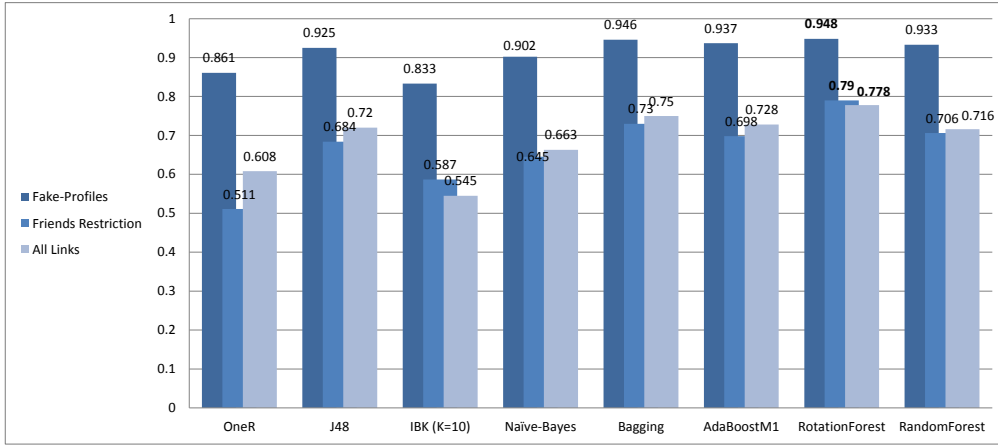


Figure 8: Classifiers AUC on the Different Datasets - it can be noted that the Rotation-Forest classifier received the highest AUC rates on all three datasets.

of view, we also calculated the Rotation-Forest classifier average users precision k , as described in Section 4.2.2, on the different datasets for different values of k ⁶.

The results of this precision at k evaluation are presented in Figure 10. It can be noticed that the Rotation-Forest classifiers presented precision at 10 of 24%, 23%, and 14% for the All links dataset, Fake profiles datasets, and Friends restriction dataset respectively. The Rotation-Forest classifier's results on the All links dataset indicated that on average 2.4 of the users' friends, which received the top ten highest probabilities of being restricted among all the friends of each user were actually restricted. However, the Rotation-Forest classifier's results on the Restricted profiles dataset indicated that on average only 1.4 of the users' friends, which received the top ten highest probabilities of being restricted among all the friends of each user had actually been restricted.

To obtain an indication of the usefulness of the various features, we also calculated the different features importance using WEKA's information gain attribute selection algorithm (see Table 4). According to the information gain selection algorithm the top two most useful features on all three datasets were the *Common-Friends* feature and the *Jaccard's-Coefficient* feature. Furthermore, according to the results there are differences between the features scores in the different datasets. For

⁶In case of the Friends profiles datasets, we calculated the average users precision for 355 SPP application users only, which for certain were familiar with alphabetical interface and used it to restrict their friends.

Table 3: Classifiers' Performance on the Different Datasets

Classifier	Measure	Fake Profiles	Friends Profiles Restriction	All Links
OneR	AUC	0.861	0.511	0.608
	F-Measure	0.867	0.531	0.616
	False-Positive	0.179	0.532	0.414
	True-Positive	0.902	0.554	0.623
J48	AUC	0.925	0.684	0.72
	F-Measure	0.885	0.668	0.659
	False-Positive	0.179	0.498	0.321
	True-Positive	0.937	0.754	0.645
IBK (K=10)	AUC	0.833	0.587	0.545
	F-Measure	0.744	0.49	0.637
	False-Positive	0.174	0.289	0.749
	True-Positive	0.696	0.419	0.817
Naive-Bayes	AUC	0.902	0.645	0.663
	F-Measure	0.833	0.678	0.287
	False-Positive	0.373	0.856	0.055
	True-Positive	0.979	0.955	0.177
Bagging	AUC	0.946	0.73	0.75
	F-Measure	0.89	0.677	0.675
	False-Positive	0.171	0.403	0.3
	True-Positive	0.938	0.717	0.662
AdaBoostM1	AUC	0.937	0.698	0.728
	F-Measure	0.882	0.645	0.657
	False-Positive	0.163	0.403	0.312
	True-Positive	0.941	0.671	0.643
Rotation-Forest	AUC	0.948	0.79	0.778
	F-Measure	0.897	0.719	0.696
	False-Positive	0.158	0.336	0.275
	True-Positive	0.941	0.75	0.681
Random-Forest	AUC	0.933	0.706	0.716
	F-Measure	0.858	0.613	0.663
	False-Positive	0.14	0.278	0.369
	True-Positive	0.857	0.565	0.679

example, the *Common-Groups-Ratio* feature received a value of 0.113 in the Fake-profile dataset and a value of only 0.004 in the Friends-restriction dataset, and the *Is-Friend-Profile-Private* received a value of 0.056 in the Friends-restriction dataset and a value of only 0.0002 in the All links dataset.

5.3 Add-on Results

The SPP Firefox Add-on was downloaded more than 1,676⁷ times between the 27th of June, 2012 and the 10th of November, 2012. During that time we succeeded in collecting data with the number of installed Facebook applications from 1,676 different Facebook users. This data was collected on 21,524 different occasions. Furthermore, we also succeeded in collecting SPP users' privacy settings of at least 67 Facebook users on 129 different occasions⁸.

⁷The SPP Add-on was available for download from several locations, such as the Firefox Add-ons website and the PrivacyProtector.net website. Due to the fact that not all locations store the number of downloads, we can only estimate the number of downloads according to our HTTP Server logs.

⁸Due to the fact that not all SPP users opened their Facebook privacy settings during this time period, and probably due to problems in parsing the different Facebook privacy settings page layouts, we succeeded in collecting the SPP users' privacy settings for only a limited number of users.

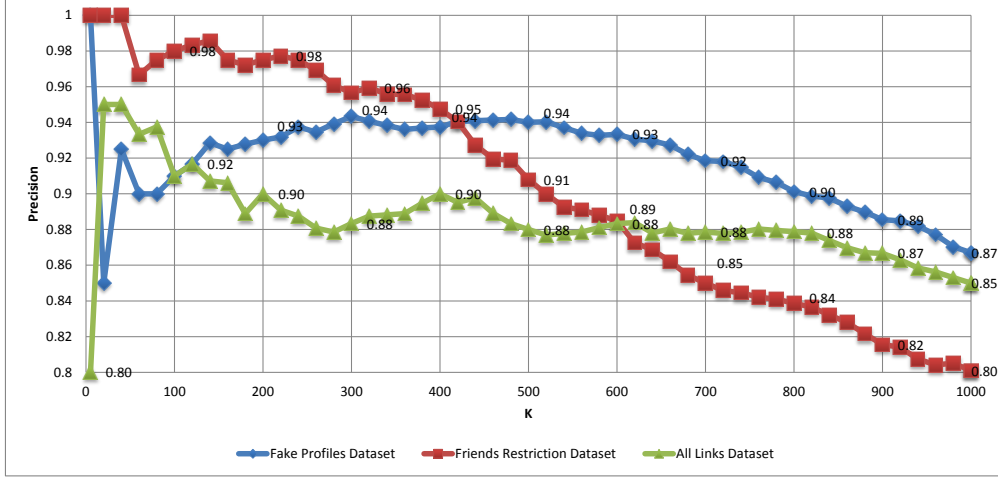


Figure 9: Rotation-Forest Precision@k - it can be seen that the classifiers' precision at 100 was 98%, 91%, and 91% for the Friends restriction dataset, Fake profiles datasets, and All links dataset respectively.

Table 4: Information Gain Values of Different Features for Different Datasets

	Common-Friends	Jaccard's-Coefficient	Common-Groups-Number	Common-Groups-Ratio	Is Friend Profile Private	Common-Posts-Ratio	Friend Friends-Number	Common-Chat-Messages	Common-Posts-Ratio	Common-Posts-Number	Chat-Message-Ratio	Tagged-Photos-Number	Common-Photos-Ratio	Tagged-Videos-Number	Common-Video-Ratio	Is Family
Fake Profiles	0.521	0.466	0.122	0.113	0.03	0.029	0.047	0.045	0.029	0.034	0.045	0.035	0.029	0.005	0.001	0
Friends Restriction	0.036	0.012	0.004	0.003	0.056	0.016	0.003	0.003	0.016	0.008	0	0.003	0.005	0.001	0.001	0
All Links	0.102	0.079	0.015	0.017	0.0002	0.015	0.01	0.008	0.015	0.014	0.005	0.008	0.008	0.001	0.001	0
Average	0.22	0.186	0.047	0.044	0.029	0.02	0.02	0.019	0.02	0.019	0.017	0.015	0.014	0.002	0.001	0

By analyzing the collected applications data we discovered that the number of Facebook applications installed on users' profiles, at the time they initially installed our Add-on, ranged from one installed application to 1,243 installed applications, with an average of 42.266 applications per user. Moreover, according to the installed applications distribution, we can observe that about 34.96% of the users have less than ten applications installed on their profiles. However, 30.31% of the users have at least 40 installed applications and 10.68% have more than 100 applications installed on their profiles (see Figure 11).

In addition to calculating the statistics on the number of installed Facebook applications, we also tested if the SPP users had used the Add-on to remove part of their installed Facebook applications. In order to identify if a user has removed the Facebook applications using the Add-on, we check what the user's installed applications numbers up to a day after the Add-on is initially installed. Our Add-on succeeded to collect the data of 626 users a day after the Add-on is initially installed. Out of these 626 users 111 (17.73%), had removed 1,792 applications, while 149 (23.8%), users added 192 applications, and 366 (58.47%) users did not add or remove any applications (see Figure 12). A closer look at the application removal data reveals that on average each user from the 111 users removed 34.7% of all installed applications and 32 (28.8%), users had removed at least 50% of all their installed applications (see Figure 13).

If we look at the overall time period of our experiments, from the 27th of June, 2012 to the 10th of November, 2012, we can see that out of 1,676 users 335 (19.99%), users decreased the number of

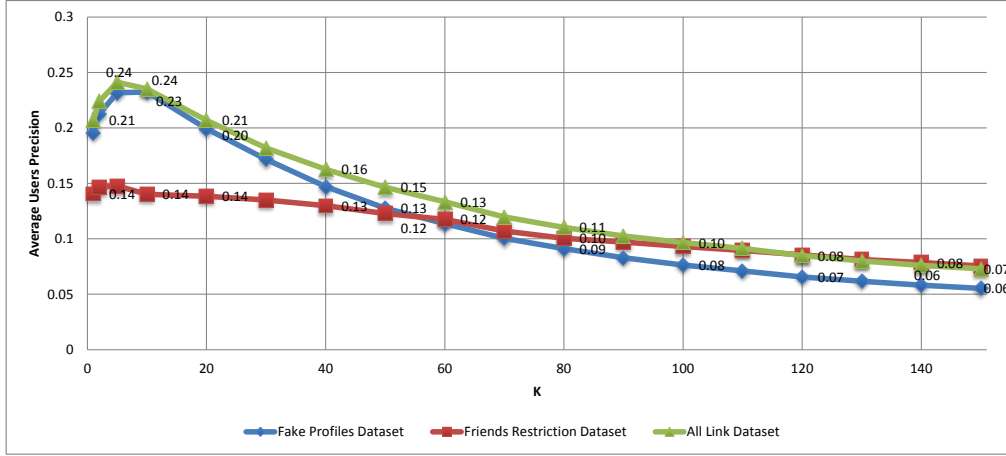


Figure 10: Rotation-Forest average users' precision@k - it can be seen that the classifiers' average users precisions at 20 were 21%, 20%, and 14% for the All links dataset, Fake profiles datasets, and Friends restriction dataset respectively.

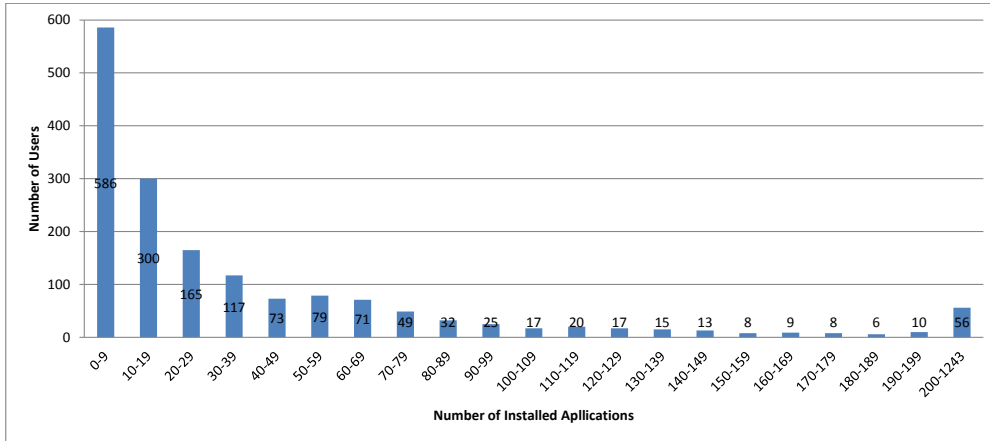


Figure 11: Distribution of The Number of Installed Facebook Applications - it can be noted that 10.68% of the users have more than 100 applications installed on their profiles.

installed applications on their profiles. These users had removed 5,537 applications with an average of 16.52 application removals per user and a median of seven.

In addition to checking how many applications were removed, we also checked how many new applications were installed to each of the Add-on users. To achieve this goal, we first focused on the group of Add-on users, which to the best of our knowledge had more applications installed on their profile at the end of the 10th of November, 2012 than in the day they first installed the Add-on. For these user groups, we calculated how many applications were added to their profiles on an average each week. We discovered that out of 1,676 users, 389 (23.2%), users increased the number of installed applications on their profile ranging from 0.05 to 107.33 average of new application installations per day (with a median of 0.636 and an average of 1.91).

We also analyzed the distribution of the privacy-settings collected from the 67 unique Add-on users (see Table 5). It can be noticed that 74.62% of the users set their default privacy settings to be exposed to everyone. Moreover, according to the users privacy settings it can be noticed that almost all the user information, except *Tag-Suggestions*, is exposed to the friends of the user. In addition, we also tested how many Add-on users changed their privacy settings during this time period, and discovered that according to our logs 14 (20.9%) Add-on users changed their privacy settings. However, after a

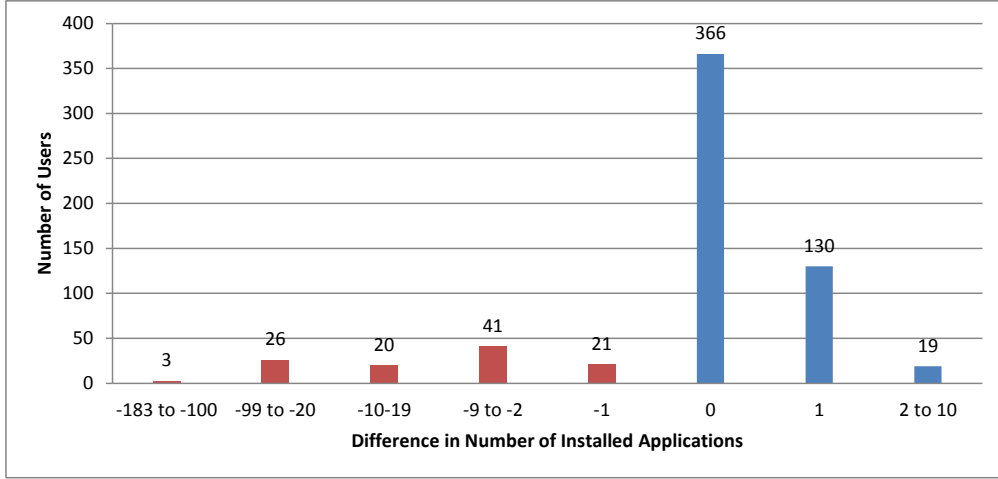


Figure 12: Distribution of the Difference in Installed Applications Number Day After the Add-on Installation.

short while the majority of these fourteen users returned to their old, less restricted privacy settings.

Table 5: Add-on Unique Users Privacy Settings Distribution

Users Settings	Default Privacy Settings	Lookup	Share Address	Send Messages	Receive Friend Req.	Tag Sugg.	View Birthday
Public/Everyone	74.62%	74.62%	-	74.63%	-	-	11.94%
Friends	23.88%	25.38%	100%	25.27%	-	52.87%	-
Friends of Friends	-	-	-	-	86.76%	-	88.06%
No one	-	-	-	-	-	23.88%	

6 Discussion

By analyzing the results presented in Section 5, we can notice the following:

First, we notice that the initial SPP application results presented relatively good performances. Although we defined the Connection-Strength heuristic to be quite simple; it presented remarkable precision, where on average 31.1% of the users' friends, which were ranked in first place were actually restricted (see Figure 6). Furthermore, the heuristic also presented an average users precision of 33.6% at 1, an average users precision of 27% at 10, and an average users precision of 11% at 100 (see Figure 7). However, the Connection-Strength heuristic was not able to present a generic method, with high true-positive rates and low false-positive rates, for recommending the restriction of links. For example, only 10% of the SPP users' friends who received a Connection-Strength with a value of 3 were restricted (see Figure 5).

Second, among all tested machine learning algorithms the Rotation-Forest classifiers performed best on all datasets, with especially good results for AUC of 0.948 and a false-positive rate of 15.8% on the Fake profiles dataset (see Table 3).

Third, according to the results, the Rotation-Forest classifiers' average users precision at 1 on the All links dataset datasets and on the Fake-profiles it was 21% and 20%, respectively. These results were worse than the results presented by the Connection-Strength heuristic, which presented an average users precision at 1 of 34%. However, the classifiers average users precision at k for higher k values was nearly the same as the Connection-Strength heuristic's precision at k . For example,

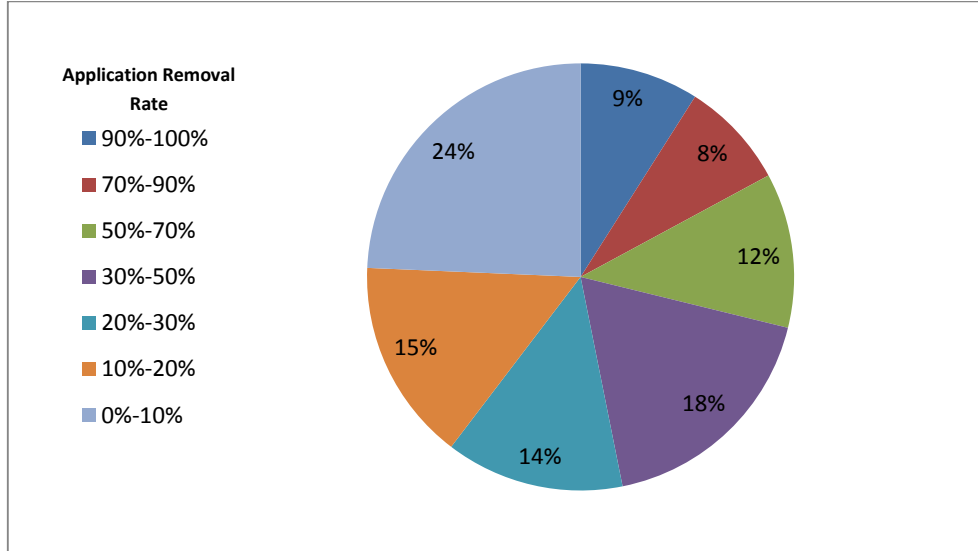


Figure 13: Removed Application Percentage - every slice of the pie represent the percent of applications removed by a percent of users. For example, 9% of the Add-on users removed between 90%-100% of all applications installed on their profiles. It can be noted that about 29% of all the users have removed more than 50% of all applications installed on their profiles.

Connection-Strength heuristic average users precision at 20 was 22%, while the Rotation-Forest classifiers' average users precision at 20 was 21% on All links dataset and 20% on Fake-profiles datasets (see Figures 7 and 10). Nevertheless, using Rotation-Forest classifiers has many advantages, which the Connection-Strength heuristic does not have, such as presenting a generic model for links restriction recommendation, and presenting the restriction probability for each link in the network without the need to compare each link to other links of the same user as is done in the case of the Connection-Strength heuristic.

Fourth, in contrast to the other classifiers, the classifier which was constructed from the Restricted friends dataset using the Rotation-Forest algorithm presented a relatively low average users precisions at 1 of 14% (see Figure 10). However, this precision is significantly better than the precision obtained by random guessing, which stands on 4.25%, in the Friends restriction dataset. We assume that this classifier presented a relatively low performance because the restricted friends in this dataset were mainly real friends, which the SPP users chose to restrict for different reasons. We assume that these reasons cannot be inferred from the features we extracted in the SPP's initial version.

Five, when the Rotation-Forest classifiers were evaluated on the general scenario of predicting which links to restrict among all users' links, the classifiers presented very high precision rates. For example, the Rotation-Forest classifier, which was constructed from Fake profiles dataset links presented 91% precision at 100 and 94% precision at 500 (see Figure 9). Moreover, the Rotation-Forest classifier, which was constructed from the Friends restriction dataset presented impressive precision at 100 of 98%. These results indicate that the classifiers can be used not only by the social network users, but also by the online social network administrator in order to identify fake profiles among all profiles on the network.

Six, according to the information gain results we can conclude that on all datasets the most useful features were the *Common-friends* feature and the *Jaccard's-Coefficient* feature (see Table 4). Additionally, the *Is-Friend-Profile-Private* was found to be very useful in the case of the Friends restriction dataset indicating that friends, which have their profile set to be private have a higher likelihood of being restricted. Moreover, according to these results it is noticeable that the *Is Family*, and the *Tagged-Video-Number* features were not so useful. Removing the extraction of these features in future versions can assist in improving the SPP application run time without significantly affecting

the results.

Seven, according to the applications statistic results it can be noted that many Facebook users installed many applications on their Facebook accounts, which can jeopardize their privacy. According to our results out of 1,676 examined Facebook users, 30.31% of the users had at least forty installed applications and 10.68% of the users had more than a hundred installed applications (See Figure 11). Moreover, according to our results, out of the 111 users, which used the SPP Add-on for application removal, 28.2% removed more than 50% of all their installed applications a day after they installed the SPP Add-on (see Figure 13). These results indicate that in many cases the installed applications are unwanted or unneeded applications. Furthermore, our results also uncovered an alarming phenomenon; namely, many Facebook users install new applications weekly. According to our results 389 out of 1,676 users had increased the number of installed Facebook applications on their profiles with an average number of 1.91 new application installations per week.

Eight, according to the collected users' privacy statistics we can see that almost all of the examined users information is available to friends, leaving the users' information exposed to fake friends (See Table 5). In addition, the vast majority of examined users also set their "Default Privacy Setting" to be accessed by everyone. This result indicates that many users do not protect their personal information and leave it exposed to the public's view.

Lastly, according to the overall results we can see that SPP software has assisted its users in protecting their privacy both by restricting friends and by removing unwanted Facebook applications. However, the SPP software did not succeeded in assisting users to improve their privacy settings in most cases.

7 Conclusions

In this study, we presented the SPP software, which aims to better protect user's privacy in Facebook. We presented in detail the general architecture of the SPP software (see Section 3). According to this architecture, the SPP software can be divided into three layers of protection. The first layer helps to restrict a user's friend's access to personal information. The second layer helps to identify and warn the user about installed Facebook applications, which can violate the user's privacy. The third layer helps the user to adjust their privacy settings with one click. According to this software architecture the heart of the SPP software lays in the function, which is responsible for recommending to each user, which friends to restrict. This function can be a simple heuristic or a more complicated machine learning classifier.

In the initial version of the SPP software we chose to implement the Connection-Strength heuristic, which was responsible for recommending to each user, which of his friends to restrict (see Section 4.1). By using the software interfaces and the Connection-Strength heuristic recommendations, 527 out of 3,017 software users had restricted 9,005 friends in less than four months. According to the results of the Connection-Strength heuristic we can conclude that the Connection-Strength heuristic presented the users with a relatively remarkable recommendation. By using these recommendations the SPP users' had restricted 30.87% of their friends, which appeared in the first position in the application's Restriction interface (see Figure 6). However, the Connection-Strength did not provide a general method of identifying which of the users' links need to be restricted.

To create general link restriction recommendation methods we chose to use a supervised learning approach. By using the unique data, which was created by the initial SPP version, we created three types of datasets for different friend restriction scenarios(see Section 4). We used these datasets to construct and to compare different machine learning algorithms to identify which algorithm can provide the best results (see Section 4.2.2). We discovered that the Rotation-Forest algorithm presented the best AUC and false-positive rates results on all three datasets (see Table 3). We then showed that the Rotation-Forest classifiers had been created from the Fake profiles dataset and the All links dataset presented good average users precision at k results (see Figure 10). Furthermore, we demonstrated that these classifiers can provide Facebook administrators with a method, which can assist them in identifying fake profiles among all users in the network (see Figure 9). However, according to our evaluations these classifiers suffer from relatively high false-positive rates. We believe that these false-

positive rates can be considerably reduced if the network administrator uses the classifiers to evaluate several links, instead of one link only, for each suspicious profile, which the classifiers had marked off as being a fake profile. We hope to verify this assumption in future research.

In this study, we also collected statistics from 1,676 different Facebook users on the number of applications installed on their Facebook profiles during different time periods. According to these statistics we discovered that many Facebook users have an alarming number of applications installed on their profiles, where 30.31% of the users had at least forty installed applications (See Figure 11). In addition, our statistical analysis also showed that many users continued to install new applications with an average of 1.91 new applications every week. Fortunately, according to our results, we discovered that at least 111 SPP Add-on users had used the SPP Add-on to improve their privacy and removed at least 1,792 applications (see Figure 12). These results indicate that by making users more aware of the existence of installed applications we can assist in reducing the number of the installed applications, and may decrease the exposure of users' personal information to third party companies. Furthermore, in this study we also collected statistics on the privacy settings of 67 unique Facebook users. According to these privacy statistics we can conclude that a majority of the users expose their private information to friends, and in many cases even to the public. Once again, these statistics sharply demonstrate how exposed Facebook users information can be to both fake profile attacks and third party Facebook applications.

In the future, we hope to continue our study and provide an updated version of the SPP Add-on, which will be able to support more web browsers, such as Chrome and Internet Explorer. In addition, we plan to remove the extraction of less useful features, like the *Tagged-Video-Number* feature, and through this improve the SPP application performance. We hope that these improvements will assist SPP users to restrict more fake profiles and through this increase the size of our classifiers' training set.

We also believe that this study has several future research directions, which can improve the identification of fake profiles in online social networks. A possible direction is to extract more complicated topological features, such as the number of communities of each user, and use them to construct better classifiers with lower false-positive rates. In our previous study [16], we demonstrated that these type of features can assist in identifying fake profiles. Another possible direction, which can be used to improve the classifiers performances is to construct the classifiers by using oversampling techniques, like SMOTE [5], to deal with the dataset imbalance issue instead of the under sampling techniques we used in this study. We also hope to test the constructed classifiers performance on different online social networks, such as Google+ and Twitter. Another future direction we want to examine is the usage of the SPP software as an educating tool. We also hope to examine if users utilizing the SPP software to restrict friends and remove applications became more aware of their privacy, and as result tended to accept less friend requests and installed fewer applications. In future studies, we also hope to perform a deeper analysis on the unique datasets we obtained with the SPP software and extract different insights on connections between Facebook users. We also hope to test the developed algorithms to improve users' security and privacy in other online social networks.

8 Availability

The Social Privacy Protector and parts of its source code are available for download from <http://www.socialprotector.net>. The Friend Analyzer Facebook application is available to download from https://apps.facebook.com/friend_analyzer_app. A video with detailed explanations on how to use the SPP application is available in <http://www.youtube.com/watch?v=Uf0LQsP4sSs>

References

- [1] Y. Altshuler, M. Fire, N. Aharony, Y. Elovici, and A. Pentland. How many makes a crowd? on the evolution of learning as a factor of community coverage. *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 43–52, 2012.

- [2] M. Anwar and P. Fong. A visualization tool for evaluating access control policies in facebook-style social network systems. 2012.
- [3] A. Bigos. New facebook app helps protect kids against online criminals. <http://www.foxcharlotte.com/news/top-stories/New-Facebook-App-Helps-Protects-Kids-Against-Online-Criminals-162540726.html>, 2012. [Online; accessed 21-September-2012].
- [4] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 93–102. ACM, 2011.
- [5] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, 2011.
- [6] D. DeBarr and H. Wechsler. Using social network analysis for spam detection. *Advances in Social Computing*, pages 62–69, 2010.
- [7] W. Defensio. <http://www.defensio.com/>. [Online; accessed 1-September-2012].
- [8] M. Egele, A. Moser, C. Kruegel, and E. Kirda. Pox: Protecting users from malicious facebook applications. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 288–294. IEEE, 2011.
- [9] Facebook. <http://www.facebook.com/>. [Online; accessed 5-October-2012].
- [10] Facebook. Facebook apis reference. <https://developers.facebook.com/docs/reference/apis/>. [Online; accessed 5-October-2012].
- [11] Facebook. Report abuse or policy violations.
- [12] I. Facebook. Quarterly report pursuant to section 13 or 15(d) of the securities exchange act of 1934. http://www.sec.gov/Archives/edgar/data/1326801/000119312512325997/d371464d10q.htm#tx371464_14, 2012.
- [13] Facebook-Newsroom. <http://www.facebook.com>. [Online; accessed 5-October-2012].
- [14] M. Fire, D. Kagan, A. Elisahr, and Y. Elovici. Social privacy protector official website. <http://socialprotector.net/>, 2012. [Online; accessed 21-September-2012].
- [15] M. Fire, D. Kagan, A. Elishar, and Y. Elovici. Social privacy protector-protecting users’ privacy in social networks. pages 46–50, 2012.
- [16] M. Fire, G. Katz, and Y. Elovici. Strangers intrusion detection-detecting spammers and fake profiles in social networks based on topology anomalies. *Human Journal*, 1(1):26–39, 2012.
- [17] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici. Link prediction in social networks using computationally efficient topological features. In *Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Confernece on Social Computing (SocialCom)*, pages 73–80. IEEE, 2011.
- [18] Google+. <https://plus.google.com/>. [Online; accessed 1-September-2012].
- [19] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM, 2004.
- [20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [21] I. Kahanda and J. Neville. Using transactional information to predict link strength in online social networks. In *Proceedings of the Third International Conference on Weblogs and Social Media (ICWSM)*, 2009.

- [22] J. Kuzma. Account creation security of social network sites. *International Journal of Applied Science and Technology*, 1(3):8–13, 2011.
- [23] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: social honeypots+ machine learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010.
- [24] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.
- [25] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.
- [26] LinkedIn. <http://www.linkedin.com/>. [Online; accessed 1-September-2012].
- [27] Y. Liu, K. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: User expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 61–70. ACM, 2011.
- [28] M. Madden and K. Zickuhr. 65% of online adults use social networking sites. <http://pewinternet.org/Reports/2011/Social-Networking-Sites.aspx>, 2011.
- [29] S. Mahmood and Y. Desmedt. Poster: preliminary analysis of google+’s privacy. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 809–812. ACM, 2011.
- [30] S. Nelson, J. Simek, and J. Foltin. The legal implications of social networking. *Regent UL Rev.*, 22:1–481, 2009.
- [31] Nielsen. The social media report. <http://blog.nielsen.com/nielsenwire/social/>, 2011.
- [32] H. Popkin. Facebook app helps ferret out pedophiles. <http://www.nbcnews.com/technology/technolog/facebook-app-helps-ferret-out-pedophiles-871761>, 2012. [Online; accessed 21-September-2012].
- [33] PrivAware. <http://apps.facebook.com/privaware/>. [Online; accessed 1-September-2012].
- [34] M. Rahman, T. Huang, H. Madhyastha, and M. Faloutsos. Efficient and scalable socware detection in online social networks. In *Proceedings of the 21st USENIX conference on Security symposium*, pages 32–32. USENIX Association, 2012.
- [35] M. Rahman, T. Huang, H. Madhyastha, and M. Faloutsos. Frappe: Detecting malicious facebook applications. 2012.
- [36] ReclaimPrivacy. <http://http://www.reclaimprivacy.org/>. [Online; accessed 1-September-2012].
- [37] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.
- [38] T. Stein, E. Chen, and K. Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, page 8. ACM, 2011.
- [39] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.
- [40] Twitter. <http://www.twitter.com/>. [Online; accessed 1-September-2012].
- [41] UnitedParents. <http://www.unitedparents.com/>. [Online; accessed 1-September-2012].

- [42] A. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.
- [43] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering social network sybils in the wild. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 259–268. ACM, 2011.
- [44] ZoneAlarm. <http://www.zonealarm.com/>. [Online; accessed 1-September-2012].